Department of Mathematics
I Semester 2020-2021
MAL 342 Analysis and Design of Algorithms
Minor Weightage 25%
Date 11.11.20 Time 1 PM -2.20 P.M

Note: You may refer to the algorithm which are standard or discussed in the class.

Q1. (a) Let f(n) be a degree k polynomial such that the coefficient of n^k is positive. Show that f(n) = $\Omega(n^k)$ using the definition of Ω (not using limit). You may assume that f(n)=$O(n^k)$. [2]

(b) Apply Master Theorem, if applicable, to express T(n) using big oh, where
T(n)= 3T(2/3)+n log n, else reason why Master Theorem will not be applicable . [2]

(b) Let $T_1(n)=7T_1(n/2)+n^2$ and $T_2(n)=aT_2(n/4)+n^2$ describe the worst case running time of algorithm 1 and algorithm 2, respectively , to solve a problem. What is the largest integer value for a such that algorithm 2 runs asymptotically faster than algorithm 1? [3]

(c) Solve the recurrence relation

$T(n)=2T(\lfloor n/2 \rfloor+17)+n.$ [3]

Q2. Let two sets A and B with m and n ($m \le n$) integers are stored in two arrays S and T, respectively. S and T are not necessarily sorted. Design an algorithm to compute $A \cup B$ and $A \cap B$ in O(n log m) time. [3]

Q3. Let S be a set of n elements. For any $x \in S$, the rank of x, r(x), is $|\{y \in S \mid y \le x\}|$. Given S in the form of an array A and two numbers m_1 and m_2, find S_1, where $S_1=\{ y \mid m_1 \le r(y) \le m_2 \}$ in O(n) time. [3]

Q4. Suppose you are a consultant for a small investment company. You are given the price per share for a stock for n consecutive days in the past. Suppose during this period, they wanted to buy 1,000 shares on some day and sell all of these shares on some (later) day. They want to know: When should they have bought and when should they have sold in order to have made as much money as possible? If there was no way to make money during the n days, you should report this instead. Design an O(n log n) time algorithm to solve this problem. Justify the claim of the complexity of the algorithm. [5]

Q5. Design a divide and conquer based algorithm to construct a binary tree given the preorder numbering P[1..n] and inorder numbering I[1..n] of the binary tree. Prove the correctness of your algorithm and analyze your algorithm. [4]
